

Bibliografía para la enseñanza de tecnologías basadas en TCP/IP, experimentando con user mode Linux

Aldo Abel Crespo - Ingeniero Electrónico, Máster en Ingeniería Electrónica
crespoa@ing.unlpam.edu.ar

Lorena G. Franco – Analista Programador, Ingeniería en sistemas
francol@ing.unlpam.edu.ar

Resumen

De la vasta literatura existente en el área de las redes de datos, la mayoría de ella, no contempla facilidades de experimentación para aquellos lectores con nulas o escasas posibilidades de acceso a una compleja infraestructura de laboratorio.

Una opción bibliográfica superadora respecto de las existentes debería contemplar una plataforma de aprendizaje que facilite el estudio y comprensión mediante experimentación de los tópicos abordados en un curso típico de Redes de Computadoras, todo ello, sin estar sujeto a las facilidades del uso de un costoso laboratorio de experimentación generalmente basado en equipamiento propietario.

En esta dirección, el presente proyecto tiene como objetivo la construcción de una novedosa alternativa bibliográfica orientada a un curso típico de redes de computadoras que trate sobre tecnologías Ethernet y la colección de protocolos TCP/IP. Para ello se utilizará una herramienta de virtualización bajo Linux, denominada User Mode Linux, o UML en forma sintética.

UML es una herramienta de virtualización bajo Linux que permite construir complejos escenarios de red. A continuación se muestra las posibilidades de experimentación que ofrece UML cuando se trata de experimentar con servicios que se ejecutan sobre TCP/IP.

Palabras clave: TCP/IP, UML, Redes.

Contexto

El proyecto “Bibliografía para la enseñanza de tecnologías basadas en TCP/IP, experimentando con User Mode Linux” corresponde al área de Arquitectura, Redes y Sistemas Operativos.

El mencionado proyecto se encuentra financiado por la Universidad Nacional de La Pampa.

1. Introducción

El hecho de que los servicios sobre TCP/IP sean distribuidos y que avancen progresivamente en una mayor complejidad al agregar nuevas funcionalidades requiere de topologías altamente pobladas para que la experimentación sea efectiva. Por ejemplo, considerando un servicio tradicional como DHCP, si lo que se busca es experimentar con toda su funcionalidad, será necesario crear algunas subredes, proveerle capacidad de *failover* al servidor principal y a los efectos de que los hosts en las distintas subredes puedan acceder al servicio será necesario configurar en los routers que vinculan las diferentes subredes la capacidad de DHCP relay. Es evidente que los recursos materiales necesarios

para lograr la topología propuesta son amplia y generalmente excede las posibilidades de un típico laboratorio de redes.

Si lo que se pretende es el dictado de un curso de administración y gestión de servicios en red, la problemática descrita en párrafos anteriores se acentúa.

Un ambiente de experimentación más efectivo debería poseer dos características inherentes:

- No estar limitado por la complejidad topológica en lo que se refiere a los recursos de hardware necesarios para llevar a cabo el experimento
- Permitir la experimentación individual para así evitar la formación de grupos y/o comisiones de trabajo

Un laboratorio de experimentación con las características descritas parecería imposible de llevar a la práctica, pero existe una forma de realizarlo y es a través del uso de herramientas de virtualización.

Considerando las múltiples opciones existentes en cuanto a herramientas de virtualización y dado a que la plataforma de sistema operativo más utilizada en la implementación de servidores sobre TCP/IP es sin lugar a dudas Linux, se utilizará *User Mode Linux* como software para la creación de máquinas y dispositivos de interconexión de redes virtuales bajo el sistema operativo Linux.

En la sección 2 se realizará una introducción a *User Mode Linux*, se contemplarán aspectos relativos a la instalación de y se abordaran la creación de uno de los dispositivos más importantes para la interconexión de redes y/o interconexión de computadoras en red. En la sección 3 se presentará una topología apta para el estudio y experimentación de un servicio bajo TCP/IP. En la sección 4 se presentará las conclusiones de este trabajo. Y por último en la sección 5 se hace referencia a los tópicos a desarrollar en el futuro.

2. User Mode Linux

User Mode Linux es una poderosa herramienta *open source* basada en Linux que permite virtualizar computadoras y redes de computadoras. Es ideal para estudiar toda la colección de protocolos TCP/IP y tiene la capacidad de implementar dispositivos de interconexión de redes que se desempeñan en las capas 1, 2 y 3 del modelo de referencia

OSI. Desarrollado por Jeff Dik, UML, permite ejecutar múltiples instancias de Linux bajo Linux [1] [3] [4].

Desde el punto de vista de usuario, cada instancia Linux creada vía UML tiene su propio *kernel* y *filesystem*, independiente del residente en el PC anfitrión. Desde el punto de vista del PC anfitrión UML se ejecuta como un proceso normal de usuario al que se le proporciona memoria virtual y opcionalmente acceso a los dispositivos del host anfitrión.

Inicialmente UML fue concebido para ser utilizado como plataforma de prueba para los desarrolladores de software en Linux y la razón es de fácil comprensión: un fallo en el código de software bajo prueba no provoca ninguna degradación en el sistema operativo del PC anfitrión. Posteriormente se visualizó un campo de aplicación mucho más extenso relacionado al área de las Redes de Computadoras. Para tal fin UML soporta la creación de interfaces virtuales de red y la creación de dispositivos tales como *TUN/TAP*, *hubs*, *switchs*, *bridges* y *routers*. Todos los dispositivos pueden conectarse entre sí a través de las interfaces virtuales de red definidas en tiempo de configuración y de este modo un PC corriente ejecutando distintas instancias de UML se transforma en una herramienta ideal para experimentar con complejos escenarios de red. En el contexto de esta publicación, el termino “experimentar” se refiere al estudio y la interacción de los protocolos TCP/IP a través de dispositivos de interconexión y de ningún modo involucra cuestiones relacionadas a medidas de desempeño de la red bajo estudio.

El ámbito de esta publicación se centrará en la creación y utilización de un switch UML como dispositivo de interconexión.

2.1. Aspectos de Instalación

Para proceder a la instalación de User Mode Linux se necesitan distintos elementos de software. Una versión del núcleo linux independiente de la del host anfitrión pues el núcleo de Linux tiene la capacidad de ejecutarse en el espacio de usuario como cualquier otro proceso de usuario. Un *filesystem* basado en alguna de las numerosas versiones existentes de Linux (igual o distinta a la existente en el host anfitrión) y las denominadas *uml utilities*.

Una vez que se dispone de los elementos de *software* mencionados, el proceso de instalación es sencillo y aunque existe abundante literatura sobre ello, por ejemplo en [1] y [2], a continuación se indican los pasos necesarios:

1. Se crea una carpeta *uml* y se descomprime el *kernel* en ella:

```
#cd /usr/local/src/  
#mkdir uml  
#tar -xjvf linux-2.6.xx.tar.bz2
```

2. Se configura el *kernel* según la arquitectura UML y se lo compila bajo la misma arquitectura:

```
#make xconfig ARCH=um  
#make linux ARCH=um
```

3. Se instalan las utilidades UML, para ello es necesario descomprimirlas en el directorio *linux* generado en el paso 1:

```
#tar -xvf uml_utilities.tar
```

De este modo, se generará una carpeta *tools* que contendrá las utilidades de UML. En este punto sólo resta compilar las *uml utilities* introduciendo el siguiente comando desde consola:

```
tools# make && make install
```

4. Por último se debe copiar el *filesystem* escogido al directorio cuya denominación depende de la versión del *kernel* utilizado.

Respecto del *filesystem*, este puede ser pre-construido y descargarse desde [8] o bien puede diseñarse a la medida del perfil de la aplicación que lo requiera y las instrucciones para ello están en [8]. En el ámbito de esta publicación se utilizó un *filesystem* pre-construido basado en la distribución Slackware versión 12.0.

Una vez realizados los pasos descritos, se esta en condiciones de ejecutar una maquina virtual con su propio *kernel* y versión de *filesystem* aisladas del *kernel* y *filesystem* del PC anfitrión. Ello significa que un usuario puede tener privilegios de *root* en una maquina virtual y carecer de idénticos privilegios en el host anfitrión.

2.2.- Ejecución de Instancias UML

Durante el arranque, UML debe montar su propio *filesystem* y para que ello ocurra es necesario indicar el dispositivo de bloque, o el UML *Block Device Driver* o dispositivo *ubd* (unidad de disco virtual). Una opción de configuración más explícita debe indicar la cantidad de memoria física suministrada a la instancia UML en el host. De esta forma la ejecución de una instancia UML desde consola inicialmente debe constar de los siguientes comandos:

```
#./linux mem=128 ubd0=root_fs
```

La directiva anterior asigna a la instancia UML 128MB de memoria *sparse* y un dispositivo de bloque *ubd0* desde donde se accederá al *filesystem* escogido.

Un factor a destacar es que cada instancia UML necesita de su propio *filesystem*. Un escenario de experimentación que por ejemplo consista de 10 máquinas virtuales, en lo relativo al espacio de disco, consumirá 10 veces el tamaño del *filesystem* individual escogido. Por ejemplo, si el tamaño del *filesystem* es de 600 MB, las 10 instancias de UML consumirán 6 GB de espacio en disco en el host anfitrión. Para evitar este inconveniente los diseñadores de UML contemplaron el uso de la técnica COW (*Copy On Write*) en que se dispone de un único *filesystem* de solo lectura, para cualquier número de instancias UML ejecutadas, y las modificaciones necesarias que una maquina virtual debería

realizar sobre el *filesystem* original se realizan en un archivo COW propio de la instancia. Ello implica un considerable ahorro de espacio en disco, puesto que cada archivo COW es *sparse*. A continuación los parámetros necesarios desde línea de comando para ejecutar dos máquinas virtuales utilizando la técnica COW:

```
#./linux mem=128 \
    ubd0=root_fs0.cow,root_fs

#./linux mem=128 \
    ubd0=root_fsl.cow,root_fs
```

Hasta el momento, las máquinas virtuales están aisladas del host anfitrión y la utilización de un *filesystem* pre-construido para UML no incluirá todos los paquetes de software para todos los fines y perfiles de aplicación de la herramienta virtual. Una forma de hacer visible el *filesystem* del host anfitrión desde una máquina UML es mediante el siguiente comando:

```
uml# mount -t none rootfs /mnt/host
```

Ahora una máquina virtual UML tiene acceso al archivo de sistema del host anfitrión y ello permitirá transferir datos entre ambos *filesystems* y aunque este es un método simple y sencillo de implementar, es limitado. Una solución más amplia contemplaría la conexión de una máquina virtual al mundo exterior y es precisamente este tópico el que se abordará en la próxima sección de este trabajo.

2.3. UML Switch

Los mecanismos hasta aquí descritos constituyen herramientas válidas para conectar instancias UML al mundo exterior. Un mecanismo que permite crear redes virtuales aisladas del mundo exterior, es a través de la utilización del demonio `uml_switch` parte de las *utilities* UML instaladas en la sección 2.1. Este método difiere de los previamente desarrollados en el sentido de que antes de proceder a la creación de las máquinas virtuales, es necesario ejecutar en el host el proceso `uml_switch` quien abrirá un socket tipo UNIX cuyo único propósito es comunicar las instancias UML mediante datagramas:

```
host# uml_switch -unix /tmp/sw1 -daemon
```

Ahora se pueden ejecutar máquinas virtuales y conectarlas al dispositivo `uml_switch`, por ejemplo, el siguiente comando, crea una máquina virtual denominada PC1 y la conecta al dispositivo `uml_switch` en el socket cuyo nombre es `sw1`:

```
host#./linux uid=PC1 /
    ubd0=root_fsl.cow,root_fs /
    eth0=daemon,,unix,/tmp/sw1.ctl
```

Un factor a destacar es que `uml_switch` puede configurarse para actuar según dos modos de operación: como *switch* Ethernet, o como *hub* Ethernet. El comando que permite configurar el dispositivo virtual en modo *hub* es:

```
host# uml_switch -hub -unix /tmp/hb1 -daemon
```

3. Experimentando con Servicios

Tal como ocurre con la mayoría de los servicios de aplicación orientados a la red bajo TCP/IP, ellos pueden ser vistos como sencillos de experimentar cuando se omite la mayor parte de su funcionalidad.

Por ejemplo, para la experimentación de un servicio de DHCP, un escenario simplista contemplaría un segmento LAN con los clientes y algún servidor de DHCP. En cambio, un buen escenario de experimentación debería requerir de algunas subredes pobladas de hosts, *routers*, un servidor de DHCP principal y uno o más servidores de DHCP de respaldo tal como lo ilustra la Figura 1.

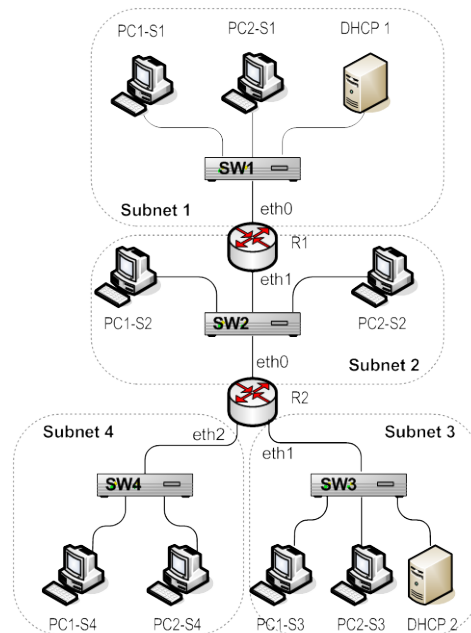


Figura 1- Ejemplo de topología construida con UML

Un escenario como el descrito por la Figura 1 permitiría incursionar en aspectos del servicio no utilizados de otro modo, pero es evidente que los recursos de hardware necesarios para llevar adelante una experimentación como la descrita requieren de numerosos componentes entre hosts y dispositivos de red. La situación se agrava aún más cuando se trata de replicar simultáneamente el experimento tal como ocurre en ambientes educativos donde por lo general se recurre a la formación de grupos y/o comisiones de alumnos.

Una alternativa válida tendiente a superar la problemática descrita es a través de la utilización de User Mode Linux que posibilita la construcción de escenarios tan o más complejos que el de la Figura 1 sin más requerimientos que el de poseer una computadora de medianos recursos.

A continuación, y a los efectos de que el lector considere un ejemplo práctico de una aplicación real, se mostrará como emplear UML para implementar la topología ilustrada en la Figura 1.

Para este caso en particular se empleará una red IP de tipo “C” dada por 192.168.100.0/24 dividida en cuatro subredes tal como se indica en la Figura 1. Cada subred debe ser dimensionada para albergar un número de hosts con los requerimientos dados en la Tabla 1:

Subred	Nº Hosts
1	14
2	30
3	90
4	10

Tabla 1- Requerimientos de Host en las subredes

Para construir la topología de la Figura 1, en primer lugar se deben crear los *switchs* Ethernet y para ello se debe ejecutar el demonio *uml_switch*. La sintaxis asociada al comando se muestra a continuación:

```
host# uml_switch -unix /tmp/sw1 -daemon
```

De este modo el *switch* que provee conectividad Ethernet en la subred 1 esta activo y ejecutándose en espacio de *background* (-daemon). El proceso debe repetirse para la creación de los *switchs* en las subredes 2, 3, y 4.

El siguiente paso comprende la ejecución de las maquinas virtuales y la conexión de sus respectivas interfaces a los *switchs* que correspondan según la Figura 1. Por ejemplo, para ejecutar las maquinas virtuales R1 y DHCP 1 la sintaxis desde línea de comando es:

```
host# ./linux umid=R1 \
    ubd0=R1.cow,root_fs \
    eth0=daemon,,unix,/tmp/sw1 \
    eth1=daemon,,unix,/tmp/sw2 \
```

```
host# ./linux umid=DHCP_1 \
    ubd0=DHCP_1.cow,root_fs \
    eth0=daemon,,unix,/tmp/sw1
```

El proceso de ejecución para el resto de las maquinas virtuales es análogo al descrito salvo las peculiaridades de cada caso (nombre de maquina virtual, nombre de archivo cow, cantidad de interfaces de red, etc).

El espacio de direcciones para cada una de las subredes se muestra en la Tabla 2 y las direcciones IP asignadas a cada interfaz de los *routers* y a los servidores se indican en la Tabla 3.

Subred	Dir. Subred	Dir. Hosts	Netmask
1	*.160	161-174	/ 28
2	*.128	129-158	/ 27
3	*.0	1-126	/ 25
4	*.192	193-206	/ 28

Tabla 2- Subredes de mascara variable

Host	Subred	Nº IP
DHCP 1	1	192.168.1.164
R1 eth0	1	192.168.1.161
R1 eth1	2	192.168.1.130
R2 eth0	2	192.168.1.129
DHCP 2	3	192.168.1.4
R2 eth1	3	192.168.1.1
R2 eth2	4	192.168.1.193

Tabla 3- Asignación de IPs

Resta configurar las interfaces de red para los dispositivos y hosts indicados en la Tabla 3. Por ejemplo la configuración de las interfaces de red para el router R1 es:

```
R1# ifconfig eth0 192.168.1.161 \
    netmask 255.255.255.240 up
R1# ifconfig eth1 192.168.1.130 \
    netmask 255.255.255.224 up
```

Una vez configuradas todas las interfaces de red de todos los dispositivos y host indicados en la Tabla 3, es necesario configurar las rutas en los *routers* R1 y R2 mediante la utilización del comando *route*. Por ejemplo, en el *router* R1 la configuración es:

```
R1# route add -net 192.168.1.160 \
    netmask 255.255.255.240 dev eth0
R1# route add -net 192.168.1.128 \
    netmask 255.255.255.224 dev eth1
R1# route add default gw 192.168.1.129
```

De manera análoga se configuran las rutas en R2 y en los hosts de las distintas subredes considerando las direcciones IP mostradas en la Tabla 3. Para que exista conectividad entre cualquiera de los hosts UML en la red se debe habilitar el *ip_forward* en ambos *routers*. Por ejemplo en R1:

```
R1#echo 1 > /proc/sys/net/ipv4/ip_forward
```

Como el propósito del trabajo trata sobre las posibilidades de experimentación ofrecida por la herramienta para la virtualización de máquinas y dispositivos de red y no sobre los aspectos relativos a la configuración de un determinado servicio, se deja al lector interesado los detalles de configuración del servicio DHCP.

4. CONCLUSION

Para lograr un aprendizaje que facilite el estudio y comprensión mediante experimentación de los tópicos abordados en un curso típico de Redes de Computadoras se requieren laboratorios cuyos elementos constitutivos no restrinjan las posibilidades de experimentación que exigen los servicios actuales. Aún cuando se disponga del equipamiento necesario para llevar adelante las prácticas de

laboratorio, es prácticamente imposible lograr una modalidad de experimentación individual y por lo general se recurre a un modo de experimentación grupal con las dificultades que ello implica debido a que los grupos y/o comisiones de experimentación en general son heterogéneos en lo que se refiere a las habilidades y capacidades constitutivas de cada integrante.

User Mode Linux es una herramienta ideal para construir complejos escenarios de red a los efectos de experimentar con protocolos y servicios de aplicación bajo Linux. Con *User Mode Linux* la experimentación individual no es una utopía y los alumnos puedan ejecutar configuraciones de laboratorio avanzadas en sus respectivas computadoras personales. Este último factor es el que ha llevado a prestigiosas universidades en el mundo a construir laboratorios de redes virtuales basados en código abierto e implementadas bajo UML [5][6][7][9][10].

5. Trabajo Futuro

Los tópicos a abordar en el futuro durante el desarrollo del proyecto abarcan temáticas tales como: experimentación con protocolos de redes de nueva generación tal como Multiprotocolo Label Switching (MPLS), calidad de servicios en redes IP (servicios integrados, servicios diferenciados), y mecanismos de control de congestión en TCP/IP.

El objetivo es avanzar y contribuir a la generación de una nueva opción bibliográfica, generando prácticas novedosas que consideren aspectos pedagógicos que permitan un mayor entendimiento de los tópicos abordados.

Formación de recursos humanos

El grupo de investigación tiene como director del proyecto al Ing. Aldo Abel Crespo. Los integrantes son Ing. Lorena Franco, Ing. Raúl Giri, Lic. Gustavo Lafuente, A.P Santiago Nicolau y A.P. Juan Carlos Hernandez.

Es objetivo del presente proyecto formar recursos humanos competentes en el área de la administración de redes y configuración de servicios en red. Desde la puesta en vigencia del proyecto se han dirigido los siguientes trabajos de tesis:

Tesis desarrolladas en el marco del presente proyecto: REDES VIRTUALES BAJO USER MODE LINUX .Tesis de la carrera Analista Programador de la Facultad de Ingeniería de la Universidad Nacional de La Pampa: Andrea Carmona, Lorena Gisela Franco. 2005.

HERRAMIENTAS DE SOFTWARE BASADAS EN LINUX PARA EL CONTROL DE ANCHO DE BANDA Y BALANCE DE CARGA. Tesis de la carrera Analista Programador de la Facultad de Ingeniería de la Universidad Nacional de La Pampa: Betiana Frank. 2007.

EXPERIMENTANDO CON IPV6 BAJO UML. Tesis de la carrera Licenciatura en Informática en la Universidad de Lujan: Martín Echeverría. Año 2009.

EXPERIMENTACIÓN DE CONTROL DE TRÁFICO Y CONTROL DE ANCHO DE BANDA EN LINUX MEDIANTE UML. Tesis de la carrera Licenciatura en Informática en la Universidad de Lujan: Gustavo Lafuente. Año 2009.

LABORATORIO DE EXPERIMENTACIÓN DE SERVICIOS DE RED BAJO TCP/IP CON UML. Tesis de la carrera Licenciatura en Informática en la Universidad de Lujan: Alberto Lucesoli. Año 2009

Bibliografía

- [1] *User Mode Linux*, By Dike Jeff, Publisher: Prentice Hall, April 12, 2006
- [2] "Virtual Networking with User-Mode Linux", Kuthonuzo Luruo, Shashank Khanvilkar. Linux For You Pro Magazine. March 2005.
- [3] J. Dike. A user-mode port of the Linux kernel. In 4th Annual Linux Showcase & Conference, Atlanta, 2000.
- [4] J. Dike. A user-mode port of the Linux kernel. In 5th Annual Linux Showcase & Conference, Oakland, California, 2001.
- [5] Web Access Exercise System (WAES) in <http://waes.tamu.edu>
- [6] NETKIT in <http://www.netkit.org/>
- [7] ADIOS Project, Version Number 7, in <http://os.cqu.edu.au/adios>
- [8] The User-mode Linux Kernel Home Page, in <http://user-mode-linux.sourceforge.net/>
- [9] OpenIPLab, in <http://openip.it.uu.se/>
- [10] Building a Virtual IPv6 Lab Using User-Mode Linux. By Salah M. S. Al-Buraiky. February 2004

Reseña biográfica

Abel Crespo es Ingeniero Electrónico y Máster en Ingeniería Electrónica de la UTFSM. Enseña asignaturas relacionadas a las Redes de Computadoras en la Facultad de Ingeniería de la Universidad Nacional de La Pampa (Argentina) y es autor de numerosas publicaciones en el área.

Lorena G. Franco es Analista Programador y estudiante en el último año de la carrera Ingeniería en Sistemas. Actualmente se desempeña como Ayudante de Cátedra en asignaturas relacionadas a las Redes de Computadoras y es Administradora del Laboratorio de Redes de la Facultad de Ingeniería de Universidad Nacional de La Pampa.